WE CLAIM:

1. A method of detecting executable code embedded in an Extensible Markup Language (XML) document, comprising:

applying Extensible Markup Language (XML) markup to a computer-generated document;

locating an executable code embedded in the document; and

applying an attribute to a root level element of the XML markup for notifying a subsequent application of the presence of the embedded executable code.

2. The method of Claim 1, further comprising:

passing the document to the subsequent application;

parsing the XML markup by the subsequent application to locate the attribute applied to the root level element of the XML markup applied to the document.

3. The method of Claim 2, whereby if the attribute is located by the subsequent application, rejecting the document as corrupted by the embedded executable code.

4. The method of Claim 2, whereby if the attribute is located, determining whether the embedded executable code associated with the attribute requires rejection of the document, and if not, continuing to parse the XML markup of the document by the subsequent application.

5. The method of Claim 2, whereby parsing the XML markup by the subsequent application includes locating the executable code embedded in the document;

determining by the subsequent application whether an attribute is present for notifying the subsequent application of the presence of the executable code; and

if the attribute is not present, then rejecting the executable code.

6. The method of Claim 2, whereby the attribute is a macros present attribute for indicating the presence of VBA code in the document.

7. The method of Claim 6, whereby the macros present attribute has a value of "yes" where an XML markup element is in the document containing the VBA code.

8. The method of Claim 7, whereby the macros present attribute has a value of "no" indicating that no XML element is in the file containing the VBA code.

9. The method of Claim 2, whereby the attribute is an embedded object present attribute indicating the presence of one or more OLE objects in the document.

10. The method of Claim 9, whereby the embedded object present attribute has a value of "yes" indicating the presence of at least one XML element in the document containing data associated with an OLE object.

11. The method of Claim 10, whereby the embedded object present attribute has a value of "no" indicating that there are no XML elements in the file containing data associated with an OLE object.

12. The method of Claim 2, whereby the attribute is an OCX present attribute indicating the presence of OCX objects in the document.

13. The method of Claim 12, whereby the OCX present object has a value of "yes" indicating that the presence of at least one XML element in the document containing data associated with an OCX object.

14. The method of Claim 13, whereby the OCX present object has a value of "no" indicating that there are no XML elements in the document containing data associated with an OCX object.

15. A method of detecting executable code embedded in an Extensible Markup Language (XML) document, comprising:

applying Extensible Markup Language (XML) markup to a computer-generated document;

locating an executable code embedded in the document;

applying an attribute to a root level element of the XML markup for notifying a subsequent application of the presence of the embedded executable code;

passing the document to the subsequent application;

parsing the XML markup by the subsequent application to locate the attribute applied to the root level element of the XML markup applied to the document; and

if the attribute is located by the subsequent application, rejecting the document as corrupted by the embedded executable code.

16. The method of Claim 15, whereby if the embedded executable code associated with the attribute does not require rejection of the document, continuing to parse the XML markup of the document by the subsequent application.

17. The method of Claim 15, whereby parsing the XML markup by the subsequent application includes locating the executable code embedded in the document;

determining by the subsequent application whether an attribute is present for notifying the subsequent application of the presence of the executable code; and

if the attribute is not present, then rejecting the executable code.

18. A computer readable medium on which is stored instructions which when executed by a computer perform a method of detecting executable code embedded in an Extensible Markup Language (XML) document, comprising:

applying Extensible Markup Language (XML) markup to a computer-generated document;

locating an executable code embedded in the document;

applying an attribute to a root level element of the XML markup for notifying a subsequent application of the presence of the embedded executable code;

passing the document to the subsequent application;

parsing the XML markup by the subsequent application to locate the attribute applied to the root level element of the XML markup applied to the document; and

if the attribute is located by the subsequent application, rejecting the document as corrupted by the embedded executable code.

19. The computer readable medium of Claim 18, whereby if the embedded executable code associated with the attribute does not require rejection of the document, continuing to parse the XML markup of the document by the subsequent application.

20. The computer readable medium of Claim 18, whereby parsing the XML markup by the subsequent application includes locating the executable code embedded in the document;

determining by the subsequent application whether an attribute is present for notifying the subsequent application of the presence of the executable code; and

if the attribute is not present, then rejecting the executable code.